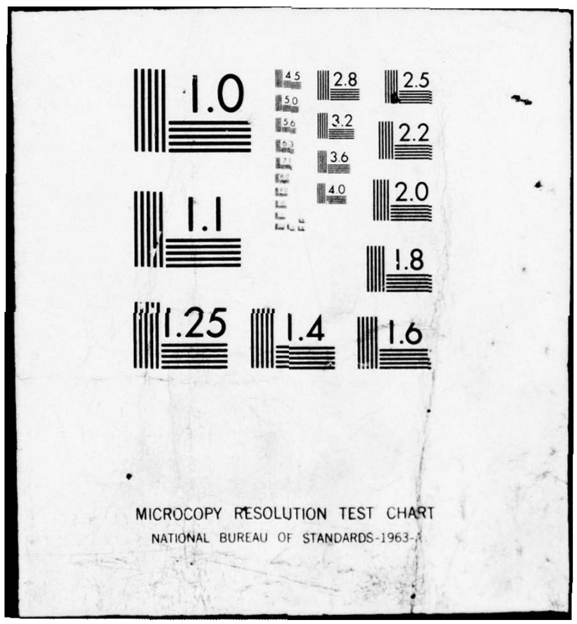
AD-A044 959 STANFORD UNIV CALIF DEPT OF OPERATIONS RESEARCH F/G 12/1
ALGORITHMIC EQUIVALENCE IN QUADRATIC PROGRAMMING I. A LEAST-DIS--ETC(U)
OCT 76 R W COTTLE, A DJANG N00014-75-C-0267
UNCLASSIFIED TR-76-26 NL

| OF |

AD
A044959

END
DATE
FILMED
11-77
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ALGORITHMIC EQUIVALENCE IN QUADRATIC PROGRAMMING I: A LEAST-DISTANCE PROGRAMMING PROBLEM

BY

RICHARD W. COTTLE and ARTHUR DJANG

TECHNICAL REPORT 76-26

OCTOBER 1976

Department of Operations Research

Stanford University

Stanford California 94305

ALGORITHMIC EQUIVALENCE IN QUADRATIC PROGRAMMING I:

A LEAST-DISTANCE PROGRAMMING PROBLEM

by

Richard W. Cottle and Arthur Djang

TECHNICAL REPORT 76-26

October 1976

DEPARTMENT OF OPERATIONS RESEARCH

Stanford University
Stanford, California

Research and reproduction of this report were partially supported by National Science Foundation Grant MCS 71-03341 ~~A04~~ and Office of Naval Research Contract N00014-75-C-0267.

ALGORITHMIC EQUIVALENCE IN QUADRATIC PROGRAMMING I:

A LEAST-DISTANCE PROGRAMMING PROBLEM

Richard W. Cottle and Arthur Djang

ABSTRACT

It is demonstrated that Wolfe's algorithm for finding the point of smallest Euclidean norm in a given convex polytope generates the same sequence of feasible points as does the van de Panne-Whinston "symmetric" algorithm applied to the associated quadratic programming problem. Furthermore, it is shown how the latter algorithm may be simplified for application to problems of this type.

1. Introduction

This is the first of a series of papers which are intended to provide a conceptual basis for the analysis and comparison of quadratic programming algorithms. The series considers a wide variety of quadratic programming problems, ranging from those having certain special structure (such as the one discussed in this paper) to more general (e.g. nonconvex) problems. With regard to these problems, the concept of equivalence between algorithms (in the sense of generating identical solution paths) is examined. Apart from their theoretical interest, the results developed in this series of papers should be of interest to those concerned with finding computationally efficient methods for solving quadratic programming problems.

The Least-Distance Problem (LDP)

In [11,12] Wolfe introduces an algorithm for finding that point of a given polytope having smallest Euclidean norm. The problem under consideration differs from least-distance problems studied by certain other authors (e.g. Tucker [7]), in that the given polytope is described as the convex hull of a given point set, rather than as the intersection of halfspaces.

In particular, let $P = \{P_1, P_2, \ldots, P_m\}$ be a given set of distinct nonzero points in E^n. The least-distance problem (LDP) can be stated as:

$$\text{minimize} \quad \|X\|^2 = X^T X$$

$$\text{subject to} \qquad X = \sum_{k=1}^{m} P_k w_k$$

$$\sum_{k=1}^{m} w_k = 1, \qquad \text{all } w_k \geq 0 .$$

As Wolfe [10] points out, problems of this type arise in applications such as the minimization of nondifferentiable functions and pattern recognition. For problems in which $m \leq n$, Wolfe [11,12] recommends the use of the "tableau variant" of his LDP algorithm. For problems with $m \gg n$, he suggests the use of other variants of the method; they follow the same path geometrically but organize the calculations in a more efficient manner.

Outline of the paper

Section 2 of the paper provides a brief description of Wolfe's LDP algorithm; it also discusses the "symmetric" quadratic programming algorithm of van de Panne and Whinston [8] as specialized to the LDP. For brevity, the latter method will be referred to as the "S-algorithm."

Section 3 shows that the structure of the LDP permits considerable simplification of the S-algorithm. It also examines the special structure of the tableaux generated by this algorithm and demonstrates that the LDP may be solved using principal pivots exclusively. Finally, the section includes a geometrical interpretation of the tableaux elements which is based on Wolfe's [11] analysis of the "tableau variant" of his LDP algorithm.

Section 4 demonstrates that the S-algorithm and Wolfe's algorithm generate the same sequence of feasible points when applied to the LDP. Furthermore, the S-algorithm and Wolfe's "tableau variant" generate the same sequence of tableaux, provided that the intermediate tableaux produced during the so-called minor cycles of each method are eliminated from consideration. Since the two methods perform an equivalent amount of work in each minor cycle, we conclude that the S-algorithm and the "tableau variant" of Wolfe's LDP algorithm are equivalent in terms of the computational effort required to solve the LDP.

The appendix of the paper is devoted to a discussion of degeneracy in the LDP, and to a summary of von Hohenbalken's method [9] for maximizing pseudoconcave functions on polytopes. When applied to the LDP, this method is identical to Wolfe's algorithm.

2. Brief Summary of the Algorithms

a. Wolfe's LDP algorithm. Initially, Wolfe [11] expresses his algorithms for the LDP in geometric language. Later, he puts the problems in a tabular format and identifes the geometric significance of the tableau entries. For the convenience of the reader, we review the notations and terminology used in [11].

Let Q denote a set of k column vectors (points) in Euclidean n-space, E^n. It will facilitate our discussion to assemble this set of vectors as an $n \times k$ matrix, also denoted Q. The affine hull of Q, is the set

$$A(Q) = \{X \in E^n : X = Qw,\ e^T w = 1\}$$

where $e^T = (1,\ldots,1) \in E^k$, $w^T = (w_1,\ldots,w_k) \in E^k$. The <u>convex hull</u> of Q is the set

$$C(Q) = \{X \in E^n : X = Qw,\ e^T w = 1,\ w \geq 0\} \ .$$

The set Q is <u>affinely independent</u> if $q \in A(Q \setminus \{q\})$ is false for all $q \in Q$. (Note that the backward slash represents set-theoretic difference and $\{q\}$ is the set with one element: q.) A point $X \in E^n \setminus \{0\}$ determines a hyperplane

$$H(X) = \{Y \in E^n : Y^T X = X^T X\}$$

passing through X and normal to the line $\overline{OX}$. The notation $O/H/S$ means that the <u>hyperplane</u> H <u>separates the set</u> S from the origin O, while the notation $O/S/H$ means that the <u>set</u> S <u>lies on the near side of the hyperplane</u> H. (Wolfe does not use these notations, but he does use the concepts they represent.) An affinely independent subset Q of P (the given set of k points) is a <u>corral</u> if the point of least norm in $C(Q)$ belongs to the relative interior of $C(Q)$. By convention, a singleton is a corral.

Wolfe's algorithm consists of a finite number of <u>major cycles</u>, each of which consists of a finite number of <u>minor cycles</u>. In the first two flow charts below the major cycles begin at step (1); the minor

cycles consist of steps (2) and (3). At the beginning of each major cycle, a corral Q and the point $X \in C(Q)$ of least norm are known. It should be noted that the solution to the problem has this form for some corral Q.

The proof that the algorithm terminates in a solution after finitely many steps is based upon the following facts:

1. Q is always affinely independent; it changes only by the deletion of single points or by the adjunction of $P_j \in P$ in step (1).

2. The number of minor cycles (if any) with a single major cycle is at most the dimension of $C(Q)$.

3. The value of the objective function $X^T X$ is reduced in each major cycle.

4. No corral can enter the algorithm more than once as the point X is uniquely determined by the corral (and by the previous fact, $X^T X$ decreases from one major cycle to the next).

The tableau variant of Wolfe's algorithm (see Flowchart 2) has the following properties:

(a) In the step immediately following the optimality test, the point P_j for adjunction to Q is that which minimizes $X^T P_i$ from among those points P_i for which $O/P_i/H(X)$.

(b) The step which finds the point of minimum norm in $A(Q)$ (see step (2)) is accomplished by a principal pivot.

b. <u>The van de Panne-Whinston S-algorithm (specialized to the LDP)</u>

We apply the S-algorithm to the following convex quadratic programming problem, which is equivalent to the LDP:

$$\text{minimize} \quad \frac{1}{2} w^T P^T P w$$
$$\text{subject to} \quad e^T w = 1$$
$$w \geq 0 \; .$$

The Kuhn-Tucker conditions in tableau form are:

	w_I	y_J	1
y_I	P^TP	$-e$	0
w_J	e^T	0	-1

with $y_I \geq 0$, $w_I \geq 0$, $y_I^T w_I = 0$, $w_J = 0$, and y_J free. Associated with the above initial tableau are index sets $I = \{1,2,\ldots,m\}$ and $J = \{m+1\}$. I is the set of indices of the basic dual variables, and J is the set of indices of the basic primal variables. As the algorithm progresses, the sets I and J will change.

The S-algorithm generates a sequence of adjacent basic solutions of the Kuhn-Tucker equations via pivot operations, each denoted $\langle$B.V.,D.V.$\rangle$. The D.V. ("driving variable") is a nonbasic variable whose value is being increased; it is chosen to be the complement* of the most negative basic

*Note: The variable w_i is said to be the <u>complement</u> of the variable y_i. In this paper, a tableau will be termed "complementary" if $y_i w_i = 0$ for all i, $1 \leq i \leq m+1$. Otherwise, if $y_i w_i > 0$ for some i, the tableau is said to be "non-complementary."

dual variable (the "distinguished variable").

The B.V. ("blocking variable") is a basic variable whose value either decreases to zero (if the B.V. happens to be one of the basic primal variables) or increases to zero (if the B.V. happens to be the distinguished variable) as a result of the increase in the D.V., and hence leaves the basis. (Basic dual variables <u>other than</u> the distinguished variable are not eligible to play the role of B.V.) The pivot operation consists of solving the equations for the new set of basic variables.

Starting from the initial tableau, the S-algorithm performs an "exchange-pivot" or 2 × 2 principal block pivot. This yields a primal feasible point which is a member of the set $P = \{P_1,\ldots,P_m\}$ having minimum norm.

Denote a typical tableau T^k generated by the S-algorithm (after the initialization step) as:

$$\begin{array}{c|cc||c|} & w_J & y_J & 1 \\ \hline y_I & M_{II} & -M_{JI}^T & q_I \\ w_J & M_{JI} & M_{JJ} & q_J \\ \hline \end{array}$$

Note that here the set J is not necessarily a singleton (as it was in the initial tableau).

Three types of pivot operations can be performed, each of which will ensure that each successive tableau generated by the method possesses the following properties:

(1) Bisymmetry: After the pivot, the new tableau is of the following form, where $\bar{M}_{I'I'}$ and $\bar{M}_{J'J'}$ are symmetric:

$$\begin{array}{c|cc||c|} & w_{I'} & y_{J'} & 1 \\ \hline y_{I'} & \bar{M}_{I'I'} & -\bar{M}^T_{J'I'} & \bar{q}_{I'} \\ w_{J'} & \bar{M}_{J'I'} & \bar{M}_{J'J'} & \bar{q}_{J'} \\ \hline \end{array}$$

(2) Positive semidefiniteness of the matrix $\bar{M}_{J'J'}$ (and also of the matrix $\bar{M}_{I'I'}$ in the case of convex quadratic programming).

(3) Primal feasibility: $\bar{q}_{J'} \geq 0$.

The S-algorithm terminates when, in addition to these properties, the tableau is also dual feasible: $\bar{y}_{I'} = \bar{q}_{I'} \geq 0$.

The following types of pivot operations may be performed:

(a) IN-PIVOT $\langle y_t, w_t \rangle$ where $t \in I$.

This 1×1 principal pivot increases $\#J$, the number of basic primal variables, by one.

(b) OUT-PIVOT $\langle w_s, y_s \rangle$ where $s \in J$.

This 1×1 principal pivot decreases $\#J$ by one.

(c) EXCHANGE-PIVOT $\langle w_s, w_t \rangle$ $\langle y_t, y_s \rangle$, $t \in I$, $s \in J$.

In this 2×2 block principal pivot, the number of basic primal variables remains the same. The operation is performed by making two 1×1 non-principal pivots. As will be shown, this type of pivot is <u>not</u> required when the algorithm is applied to the LDP. (In fact, even the initialization step, which is nominally accomplished by an exchange-pivot, may be carried out using 1×1 principal pivots only.)

Convergence of the S-algorithm for convex quadratic programming problems is established in the following manner (van de Panne and Whinston [8]).

If some basic dual variable y_t $(t \in I)$ in the current complementary tableau is negative, the tableau does not represent a minimum. The complement of y_t, namely w_t, is a nonbasic primal variable whose value is then increased. One of the three types of pivot operations will be performed.

If the pivot is of type (a) or (c), we get a new primal feasible complementary solution, for which $f(w) = w^T P^T P w = X^T X$ has reduced value. However, if the pivot is of type (b) (an out-pivot $\langle w_s, y_s \rangle$ where $s \in J$), the driving variable w_t will be nonbasic in the new tableau, but will have positive value--the value at which its increase was blocked by w_s decreasing to zero. Thus the new tableau is "non-complementary" in the sense that the variables w_t and y_t both have nonzero values. However, there can only be m such pivots of type (b) in succession, since each step reduces the cardinality of J (denoted by $\#J$) by one, and $0 \leq \#J \leq m$. Eventually, the algorithm must generate a new complementary solution, thus completing a "major cycle"; the value of $f(w)$ is then strictly lower than its value at the beginning of the major cycle.

Assuming that each basic solution is nondegenerate, we can assert that since there exist a finite number of complementary bases, and since in each major cycle the value of $f(w)$ is reduced, no basis can repeat--hence the algorithm will terminate with the solution in a finite number of steps. (Appendix I of this paper demonstrates that the assumption of

nondegeneracy can actually be dropped for the purpose of proving that the S-algorithm will converge to the solution of the LDP.)

c. Flow charts of the algorithms. To facilitate future reference and discussion we have represented Wolfe's LDP algorithm and the van de Panne-Whinston S-algorithm in flowchart form. Wolfe's algorithm has been given in both geometric and algebraic (tableau) forms.

It is hoped that the reader will notice the flowcharts have precisely the same structure. This property warrants a comment. The van de Panne-Whinston symmetric algorithm for convex quadratic programming involves the use of 1×1 diagonal pivots and 2×2 block diagonal pivots. One of the contributions of this paper is a demonstration that only diagonal pivots are necessary when the van de Panne-Whinston S-algorithm is specialized to the LDP problem. This fact is already incorporated in the corresponding flowchart and is indicated there by the comment in the lower right-hand corner of the page. To a great extent, it accounts for the duplication of the flowchart structures.

GEOMETRIC FORM OF WOLFE'S LDP ALGORITHM

INITIALIZATION

(0) Find a point $P_i \in P$ of minimal norm.

Let X be that point and $Q = \{X\}$.

START MAJOR CYCLE with a corral Q and a point $X \in C(Q)$ of minimal norm

OPTIMALITY TEST

(1) If $X = 0$ or $0/H(X)/P$, stop: X is optimal.

Otherwise choose P_j such that $0/P_j/H(X)$ and adjoin P_j to Q.

(2) Find Y, the point of smallest norm in $A(Q)$.

This is done by solving the system

$$\left\{ \begin{array}{ll} Q^T Qy + e\lambda & = 0 \\ e^T y & = 1 \end{array} \right\}$$

and setting $Y = Qy$.

FEASIBILITY TEST

$Y \in$ rel int $C(Q)$?

Yes: Replace X by Y.

No

(3) Find Z, the point nearest to Y on the line segment $C(Q) \cap \overline{XY}$.

Delete from Q one of the points not on the face of $C(Q)$ in which Z lies.

Replace X by Z.

TABLEAU VARIANT OF WOLFE'S LDP ALGORITHM

Start with the $(m+1) \times (m+1)$ tableau $T^0 = \boxed{\begin{matrix} 0 & e^T \\ e & P^TP \end{matrix}}$

Number the rows and columns $\{0,1,\ldots,m\}$.

INITIALIZATION

(0) Let $i = \hat{i}$ minimize $T^0(i,i)$ for $i > 0$.

Set $J = \{i\}$ and let w be the vector in E^n such that

$$w_i = \begin{cases} 1 & \text{for } i = \hat{i} \\ 0 & \text{otherwise} \end{cases}$$

Pivot in T^0 on $(\hat{i},\hat{i})$ and then on $(0,0)$, obtaining T.

START MAJOR CYCLE with an index set J, vector w, and tableau T.

Let $I_R = \{1,\ldots,m\}/J$.

OPTIMALITY TEST

(1) If $T(0,i) \leq 0$ for all $i \in I_R$, stop: $X = Pw$ is optimal.

Otherwise, choose $i \in I_K$ such that $T(0,i)$ is maximal.

Replace J by $J \cup \{i\}$.

(2) Pivot in T on (i,i)

Let y be the m-vector: $y_j = \begin{cases} T(0,j) & \text{for } j \in J \\ 0 & \text{for } j \in I_R \end{cases}$

FEASIBILITY TEST

$Y_j > 0$ for all $j \in J$?

Yes: Set $w = y$ (return to Let $I_R = \{1,\ldots,m\}/J$).

No:

(3) Let $\bar{\theta} = \min \left\{ \dfrac{w_j}{w_j - y_j} : w_j - y_j > 0 \right\}$.

Let i be an index for which this minimum is attained.

Replace w by $(1-\bar{\theta})w + \bar{\theta}y$.

Replace J by $J/\{i\}$.

VAN DE PANNE-WHINSTON S-ALGORITHM SPECIALIZED TO THE LDP

Start with the tableau

Initially, $I = \{1,2,\ldots,m\}$ and $J = \{m+1\}$.

	w_I	y_J	1
$y_I =$	P^TP	$-e$	0
$w_J =$	e^T	0	-1

Denote the constant column of q, and the rest of the tableau by M.

INITIALIZATION

(0) Let $\hat{i}$ denote the index of the smallest element of the main diagonal of P^TP.

EXCHANGE-PIVOT $\langle w_{m+1}, w_{\hat{i}} \rangle$, $\langle y_{\hat{i}}, y_{n+1} \rangle$

START MAJOR CYCLE. The variables $w_j > 0$, $j \in J$, and $w_i = 0$, $i \in I$, are the barycentric coordinates of the current feasible point X with respect to P.

OPTIMALITY TEST

Let $q_t = \min_{i \in I} \{q_i\}$. If $q_t \geq 0$, stop: $X = Pw$ is optimal.

Otherwise, increase the driving variable w_t.

(2) Ratio test: Find $s \in J$ such that $q_s/m_{st} = \max_{j \in J} \{q_j/m_{jt} : m_{jt} < 0\}$

If $m_{tt} > 0$, compare the values of q_s/m_{st} and q_t/m_{tt}.

(For the LDP, we always have $m_{tt} > 0$; see Theorem 3.6.)

FEASIBILITY TEST

y_t blocks w_t? (i.e. is $m_{tt} \geq 0$ and $q_t/m_{tt} \geq q_s/m_{st}$?)

Yes: IN-PIVOT $\langle y_t, w_t \rangle$. Transfer the index t from I to J. (Return to OPTIMALITY TEST.)

No:

(3) w_s blocks the increase of w_t.

If $m_{ss} > 0$, perform the

OUT-PIVOT $\langle w_s, y_s \rangle$

Transfer the index s from J to I.

Continue the increase of the driving variable w_t. (Return to (2).)

Note: If $m_{ss} = 0$, the exchange-pivot $\langle w_s, w_t \rangle, \langle y_t, y_s \rangle$ must be performed. However, for the LDP we will always have $m_{ss} > 0$ (see Theorem 3.6).

Example of an LDP Problem (due to Wolfe [11])

Problem: minimize $X^T X$

$$\text{subject to } X = \sum_{k=1}^{3} P_k w_k, \quad \sum_{k=1}^{3} w_k = 1, \; w_k \geq 0$$

$$\text{where } P = [P_1 \quad P_2 \quad P_3] = \begin{bmatrix} 0 & 3 & -2 \\ 2 & 0 & 1 \end{bmatrix} .$$

The S-algorithm, applied to this problem, generates the following sequence of tableaux:

	w_1	w_2	w_3	y_4	1
y_1	4	0	2	-1	0
y_2	0	9	-6	-1	0
y_3	2	-6	5	-1	0
w_4	1	1	1	0	-1

The initial point having barycentric coordinates $w = (0,0,0)$ with respect to P is not feasible since $e^T w \neq 1$. Apply initialization step.

INITIALIZE: EXCHANGE-PIVOT $\langle w_4, w_1 \rangle$, $\langle y_1, y_4 \rangle$

	w_2	w_3	w_4	y_1	1
y_2	13	-4	-4	1	-4
y_3	-4	5	-2	1	-2
y_4	-4	-2	4	-1	4
w_1	-1	-1	1	0	1

Tableau corresponds to the primal feasible vertex P_1 having barycentric coordinates $w = (1,0,0)$. Increase w_2, since y_2 is distinguished.

IN-PIVOT $\langle y_2, w_2 \rangle$

$$\begin{array}{c} \\ y_3 \\ y_4 \\ w_1 \\ w_2 \end{array} \quad \begin{array}{c} \\ \\ = \frac{1}{13} \\ \\ \\ \end{array} \begin{array}{cc|cc||c} w_3 & w_4 & y_1 & y_2 & 1 \\ \hline 49 & -42 & 17 & -4 & -42 \\ -42 & 36 & -9 & -4 & 36 \\ \hline -17 & 9 & 1 & -1 & 9 \\ 4 & 4 & -1 & 1 & 4 \end{array}$$

Tableau corresponds to the point R with $w = (9/13, 4/13, 0)$ since y_3 is distinguished, an attempt is made to increase w_3--but w_1 blocks.

OUT-PIVOT $\langle w_1, y_1 \rangle$

$$\begin{array}{c} \\ y_1 \\ y_3 \\ y_4 \\ w_2 \end{array} \quad \begin{array}{c} \\ \\ = \\ \\ \\ \end{array} \begin{array}{ccc|c||c} w_1 & w_3 & w_4 & y_2 & 1 \\ \hline 13 & 17 & -9 & 1 & -9 \\ 17 & 26 & -15 & 1 & -15 \\ -9 & -15 & 9 & -1 & 9 \\ \hline -1 & -1 & 1 & 0 & 1 \end{array}$$

After the out-pivot, w_3, though nonbasic, has the value 9/17 (at which its increase was blocked in the previous tableau). Hence this tableau corresponds to the point S with $w = (0, 8/17, 9/17)$. Continue the increase of w_3.

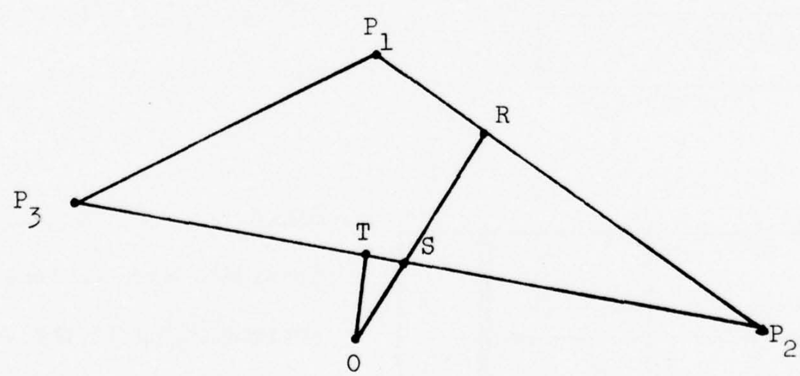
IN-PIVOT $\langle y_3, w_3 \rangle$

$$\begin{array}{c} \\ y_1 \\ y_4 \\ w_2 \\ w_3 \end{array} \quad \begin{array}{c} \\ \\ = \frac{1}{26} \\ \\ \\ \end{array} \begin{array}{cc|cc||c} w_1 & w_4 & y_2 & y_3 & 1 \\ \hline 49 & 21 & 9 & 17 & 21 \\ 21 & 9 & -11 & -15 & 9 \\ \hline -9 & 11 & 1 & -1 & 11 \\ -17 & 15 & -1 & 1 & 15 \end{array}$$

Since $y_i \geq 0$ for $i \in I$, terminate with tableau corresponding to the solution T, having $w = (0, 11/26, 15/26)$.

Wolfe's LDP algorithm, applied to the same problem, generates the following sequence of points:

Point X	Coordinates of X in E^2	Value of X^TX	Barycentric coordinates w of X with respect to P
P_1	$(0, 2)$	4	$(1, 0, 0)$
R	$(\frac{12}{13}, \frac{18}{13})$	$\frac{36}{13}$	$(\frac{9}{13}, \frac{4}{13}, 0)$
O	$(0, 0)$	0	$(-\frac{3}{7}, \frac{4}{7}, \frac{6}{7})$
S	$(\frac{6}{17}, \frac{9}{17})$	$\frac{117}{289}$	$(0, \frac{8}{17}, \frac{9}{17})$
T	$(\frac{3}{26}, \frac{15}{26})$	$\frac{9}{26}$	$(0, \frac{11}{26}, \frac{15}{26})$



3. Analysis of the S-algorithm applied to the LDP

a. Initialization

This section examines the structure of the tableau obtained by the initialization step of the S-algorithm. The tableau is shown to correspond to a point of P having minimum norm. In order to develop these results, we need to introduce the following concept.

Consider a linear system of the form $w = Mz + q$.

Definition. The tableau representing the system is said to possess the duplicate column property with respect to the variable z_i if it can be rearranged into the form:

$$\begin{array}{c|c|c|c|} & z_i & \bar{z} & 1 \\ \cline{2-4} w & q & \bar{M} & q \\ \cline{2-4} \end{array}$$

Lemma 3.0. Consider any tableau having the duplicate column property with respect to z_i. Assume that some nonsingular submatrix of $\bar{M}$ exists, and that a block pivot is performed on this submatrix. Then the resulting tableau also has the duplicate column property with respect to the variable z_i.

Proof: The result follows immediately upon partitioning the matrix $\bar{M}$ and the vectors w and $\bar{z}$ in an appropriate manner and performing the block pivot. ∎

We now examine the S-algorithm as applied to the LDP:

$$\text{minimize} \quad \frac{1}{2} w^T P^T P w$$

$$\text{subject to} \quad e^T w = 1, \quad w \geq 0 .$$

The initial tableau is as shown in Section 2b of this paper. Recall that the variable w_{m+1} is defined by $w_{m+1} = \sum_{i=1}^{m} w_i - 1$, and that $\hat{i}$ is the index of the smallest element on the diagonal of $P^T P$. (The S-algorithm resolves ties for the choice of this index arbitrarily.) The following result describes the effect of the initialization step.

Proposition 3.1. Assume that the S-algorithm is applied to the LDP.

(1) As a result of the initial exchange-pivot $\langle w_{m+1}, w_{\hat{i}} \rangle$, $\langle y_{\hat{i}}, y_{m+1} \rangle$, $(\hat{i} \in I)$, a tableau is obtained which has the duplicate column property with respect to w_{m+1}.

(2) This tableau corresponds to a point $P_{\hat{i}}$ of P having minimum norm.

(3) All subsequent tableaux generated by the S-algorithm also possess the duplicate column property with respect to w_{m+1}. The variable, w_{m+1}, having left the basis in the initialization step, never returns.

Proof: The starting tableau, T^0, is:

$$\begin{array}{c|ccc||c|}
 & w_1 & \hat{w} & y_{m+1} & 1 \\ \cline{2-5}
y_1 & m_{11} & M_1^T & -1 & 0 \\
\hat{y} & M_1 & M_{22} & -e & 0 \\
w_{m+1} & 1 & e^T & 0 & -1 \\ \cline{2-5}
\end{array}$$

Assume that m_{11} is the smallest element on the main diagonal of

$$P^TP \quad \begin{bmatrix} m_{11} & M_1^T \\ M_1 & M_{22} \end{bmatrix}$$

The initialization step calls for an exchange-pivot $\langle w_{m+1}, w_1 \rangle$, $\langle y_1, y_{m+1} \rangle$, which is of course equivalent to a block principal pivot on the nonsingular submatrix

$$\begin{bmatrix} m_{11} & -1 \\ 1 & 0 \end{bmatrix}$$

As a result of this pivot, a new tableau, T^1, is obtained. It is easily verified that this tableau has the duplicate column property with respect to w_{m+1}. Furthermore, it corresponds to a primal feasible point P_1 having coordinates $\hat{w} = (w_1, w_2, \ldots, w_m) = (1, 0, \ldots, 0)$ with respect to the set P. By our choice of m_{11} it follows that P_1 is a point of P of minimum norm. Since P_1 is feasible, the initialization step of the algorithm has been completed. (In the more general case of convex quadratic programming, the S-algorithm may require not one but several exchange-pivots to locate a primal feasible point.)

Starting from tableau T^1, the S-algorithm will perform a sequence of principal pivots. Lemma 3.0 implies that all subsequent tableaux generated by the S-algorithm will have the duplicate column property with respect to w_{m+1}. Of course, this holds true only if the variable w_{m+1} never becomes basic after the initialization.

To show that the latter condition is satisfied, observe that any tableau generated by the S-algorithm corresponds to a primal feasible point (i.e., $w \geq 0$, and $\sum_{i=1}^{m} w_i = 1$). Hence in each such tableau, we must have $w_{m+1} = \sum_{i=1}^{m} w_i - 1 = 0$. By definition of the vector y, $ey_{m+1} = P^T Pw - y$. Applying the conditions of primal feasibility and complementarity,*

$$y_{m+1} = (w^T e) y_{m+1} = w^T P^T Pw - w^T y = w^T P^T Pw = X^T X > 0 \quad .$$

We conclude that, in any tableau generated by the S-algorithm, y_{m+1} must be basic and its complement w_{m+1} must be nonbasic. ▮

Remarks.

1. The initializing block pivot on

$$\begin{bmatrix} m_{11} & -1 \\ 1 & 0 \end{bmatrix}$$

may be interpreted as a restriction of the quadratic form to the linear manifold determined by the constraints $y_1 = 0$ and $w_{m+1} = 0$ (i.e. $e^T w = 1$). This yields a primal feasible basic solution with $w_1 > 0$.

---

*Note. A similar argument is employed for the case of "noncomplementary" tableaux.

2. The w_{m+1}-row and the y_{m+1}-column are (but for a sign change) precisely the "O-row" and "O-column" described in Wolfe's statement of his algorithm [11]. Wolfe points out that "once having been pivoted in, the [0th row and column] are not used again for pivot choices."

3. In proceeding from T^0 to T^1, the pivot sequence $\langle y_1, w_1\rangle$, $\langle w_{m+1}, y_{m+1}\rangle$ could also have been used, since $m_{11} > 0$ and since the appropriate pivot element of the intermediate tableau will be positive. This corresponds to the pivot sequence employed in Wolfe's method.

b. Structure of the tableaux generated by the S-algorithm

The main result of this section is a demonstration that when applied to the LDP, the S-algorithm performs only 1×1 principal pivots; no "exchange-pivots" are necessary. This result is based on certain algebraic properties of tableaux under principal pivot operations, some of which are reviewed below.

Definition. Given any square matrix M, the nullity of M is defined by

$$n(M) = \text{order } M - \text{rank } M .$$

Now consider the homogeneous linear system represented by the tableau:

$$\begin{array}{c|cc|} & w_I & y_J \\ \hline y_I & M_{II} & M_{IJ} \\ w_J & M_{JI} & M_{JJ} \\ \hline \end{array}$$

The following result, due to Keller, shows that the nullities of the blocks M_{II} and M_{JJ} are invariant under principal transformations (i.e., sequences of principal pivots and principal rearrangements).

Theorem 3.2 (Keller [4], p. 22). Suppose that a principal transformation is performed on the tableau shown above. Let the new tableua resulting from this transformation be represented as:

$$
\begin{array}{c|cc|}
 & w_{I'} & y_{J'} \\
\hline
y_{I'} & \bar{M}_{I'I'} & \bar{M}_{I'J'} \\
w_{J'} & \bar{M}_{J'I'} & \bar{M}_{J'J'} \\
\hline
\end{array}
$$

Then

$$n(\bar{M}_{I'I'}) = n(M_{II})$$

and

$$n(\bar{M}_{J'J'}) = n(M_{JJ}) \; .$$

Now, suppose that the S-algorithm is applied to the LDP. The original tableau, T^0, may be written:

$$
\begin{array}{cc|c|c||c|}
 & & w_1 \cdots w_m & y_{m+1} & 1 \\
\hline
I_0 & \left\{ \begin{array}{c} y_1 \\ \vdots \\ y_m \end{array} \right. & P^TP & -e & 0 \\
\hline
J_0 & \{ \; w_{m+1} & e^T & 0 & -1 \\
\hline
\end{array}
$$

Associated with this initial tableau are index sets $I_0 = \{1,2,\ldots,m\}$ and $J_0 = \{m+1\}$. Let

$$M^0 = \begin{bmatrix} P^TP & -e \\ e^T & 0 \end{bmatrix}$$

Observe that $M^0_{J^0J^0} = 0$, $\operatorname{rank}(M^0_{J^0J^0}) = 0$, and $\#J^0 = 1$. It follows that $n(M^0_{J^0J^0}) = 1$. Similarly, we have $M^0_{I^0I^0} = P^TP$, $\operatorname{rank} M^0_{I^0I^0} = \operatorname{rank} P^TP = \operatorname{rank} P$, and $\#I_0 = m$. Hence $n(M^0_{I^0I^0}) = m - \operatorname{rank} P$. These observations enable us to prove the following result.

<u>Corollary 3.2.1</u>. Let Tableau T^0 be the initial tableau for the LDP. Let T^k be any subsequent tableau generated by the S-algorithm, and suppose it is represented as:

	w_I	y_J	1
y_I	M_{II}	M_{IJ}	q_I
w_J	M_{JI}	M_{JJ}	q_J

Then Tableau T^k has the following properties:

(1) *The matrix*

$$M^k = \begin{bmatrix} M_{II} & M_{IJ} \\ M_{JI} & M_{JJ} \end{bmatrix}$$

is bisymmetric, i.e. the blocks M_{II} and M_{JJ} are symmetric, and $M_{IJ} = -M_{JI}^T$. Furthermore, M^k is positive semidefinite.

(2) nullity $(M_{JJ}) = 1$

(3) $\#J = \text{rank } M_{JJ} + 1$

(4) $\det M_{JJ} = 0$

(5) M_{JJ} is symmetric and positive semidefinite

(6) If $\#J = 1$, then $M_{JJ} = 0$ (scalar).

Proof:

(1) The matrix M^k is a principal transform of the original matrix M^0, i.e. it is obtained from M^0 by a sequence of principal pivots and principal rearrangements. Observe that M^0 is a bisymmetric matrix and that it is positive semidefinite since P^TP is. By results of Cottle and Dantzig [2] and Tucker and Wolfe (cited in Parsons [6]), the properties of positive semidefiniteness and bisymmetry are preserved under principal transformations.

(2) In the initial tableau T^0, we have $n(M^0_{J^0J^0}) = 1$. By Theorem 3.2, $n(M_{JJ}) = 1$.

(3) By the definition of nullity and the fact that $n(M_{JJ}) = 1$, we must have $\#J = \text{rank } M_{JJ} + 1$.

(4) By part (3), rank M_{JJ} = order $M_{JJ} - 1$. Hence M_{JJ} is singular and $\det M_{JJ} = 0$.

(5) This follows from the fact that M_{JJ} is a principal submatrix of the bisymmetric and positive semidefinite matrix M^k.

(6) If $\#J = 1$, we must have rank $M_{JJ} = 0$ by part (3) above. Since the order of M_{JJ} is 1, it follows that $M_{JJ} = 0$. ∎

The following Lemma and Corollaries describe the structure of the tableaux in greater detail.

<u>Lemma 3.3</u>. Let

$$M = \begin{pmatrix} R^TR & R^TQ & -\hat{e} \\ Q^TR & Q^TR & -e \\ \hat{e}^T & e^T & 0 \end{pmatrix} = \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix}$$

where $\hat{e}$ and e are both column vectors of 1's, though possibly of different dimensions. Suppose

$$N = \begin{pmatrix} Q^TQ & -e \\ e^T & 0 \end{pmatrix} = \begin{pmatrix} M_{23} & M_{23} \\ M_{32} & M_{33} \end{pmatrix}$$

is nonsingular. Let

$$\bar{M} = \begin{pmatrix} \bar{M}_{11} & \bar{M}_{12} & \bar{M}_{13} \\ \bar{M}_{21} & \bar{M}_{22} & \bar{M}_{23} \\ \bar{M}_{31} & \bar{M}_{32} & \bar{M}_{33} \end{pmatrix}$$

be the principal transform of M obtained from a block pivot on the submatrix N. Then

(1) $e^T\bar{M}_{21} = -\hat{e}^T$

(2) $e^T\bar{M}_{22} = 0^T$

(3) $e^T\bar{M}_{23} = 1$.

Proof. We have

$$\begin{pmatrix} Q^TQ & -e \\ e^T & 0 \end{pmatrix}\begin{pmatrix} \bar{M}_{22} & \bar{M}_{23} \\ \bar{M}_{32} & \bar{M}_{33} \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0^T & 1 \end{pmatrix}$$

Therefore:

(4) $Q^TQ\,\bar{M}_{22} - e\bar{M}_{32} = I$

(5) $Q^TQ\,\bar{M}_{23} - e\bar{M}_{33} = 0$

(6) $e^T\,\bar{M}_{22} + 0\,\bar{M}_{32} = 0^T$

(7) $e^T\,\bar{M}_{23} + 0\,\bar{M}_{33} = 1.$

Equations (6) and (7) are just (2) and (3). To verify (1), we compute:

$$\begin{pmatrix} \bar{M}_{21} \\ \bar{M}_{31} \end{pmatrix} = -\begin{pmatrix} \bar{M}_{22} & \bar{M}_{23} \\ \bar{M}_{32} & \bar{M}_{33} \end{pmatrix}\begin{pmatrix} Q^TR \\ \hat{e}^T \end{pmatrix} = \begin{pmatrix} -\bar{M}_{22}Q^TR - \bar{M}_{23}\hat{e}^T \\ -\bar{M}_{32}Q^TR - \bar{M}_{33}\hat{e}^T \end{pmatrix}$$

From (2) and (3), we have

$$e^T\bar{M}_{21} = -\,(e^T\bar{M}_{22}Q^TR) + e^T\bar{M}_{23}\hat{e}^T) = -\hat{e}^T ,$$

which is (1). ∎

Corollary 3.3.1. Let the tableau T^k be obtained from the initial tableau T^0 by a principal transformation. Suppose that T^k is written as:

$$
\begin{array}{cc|cc|c||c|}
 & & w_{I_R} & w_{m+1} & y_J & 1 \\
\hline
I \left\{ \begin{array}{c} \\ \\ \end{array} \right. & \begin{array}{c} y_{I_R} \\ y_{m+1} \end{array} & \begin{array}{c} M_{I_R I_R} \\ M^T_{I_R,m+1} \end{array} & \begin{array}{c} M_{I_R,m+1} \\ M_{m+1,m+1} \end{array} & \begin{array}{c} -M^T_{J,I_R} \\ -M^T_{J,m+1} \end{array} & \begin{array}{c} M_{I_R,m+1} \\ M_{m+1,m+1} \end{array} \\
\hline
J \{ & w_J & M_{J,I_R} & M_{J,m+1} & M_{JJ} & M_{J,m+1} \\
\hline
\end{array}
$$

where the index set I_R is defined as $I_R = I \setminus \{m+1\}$, $\#I_R = r$, $\#I = r+1$, and $\#J = \ell = m-r$. Let $e = (1,1,\ldots,1)^T \in E^{\ell}$ and $\hat{e} = (1,1,\ldots,1)^T \in E^r$. Then the following properties hold:

$$(1) \quad e^T M_{J,I_R} = -\hat{e}^T$$

$$(2) \quad e^T M_{JJ} = 0^T$$

$$(3) \quad e^T M_{J,m+1} = 1 .$$

Proof: The initial tableau T^0 may be written:

$$
\begin{array}{c|cc|c||c|}
 & w_{I_R} & w_J & y_{m+1} & 1 \\
\hline
y_{I_R} & R^T R & R^T Q & -\hat{e} & 0 \\
y_J & Q^T R & Q^T Q & -e & 0 \\
\hline
w_{m+1} & \hat{e}^T & e^T & 0 & -1 \\
\hline
\end{array}
$$

where $I_R = \{1,2,\ldots,r\}$, $J = \{r+1, r+2, \ldots, m\}$, $R = [P_1 \;\; P_2 \;\; \cdots \;\; P_r]$ and $Q = [P_{r+1} \;.\; P_{r+2} \;\; \cdots \;\; P_m]$.

To obtain tableau T^k, a block pivot on the matrix

$$\begin{pmatrix} Q^TQ & -e \\ e^T & 0 \end{pmatrix}$$

must be performed. The required result follows directly from Lemma 3.3. ∎

The following result is a special case of the preceding Corollary.

Corollary 3.3.2. Suppose that some tableau T^k generated by the S-algorithm possesses the property that $\#J = 1$. Then the tableau must have the following structure:

$$\begin{array}{cc|cc|c||c|}
 & & w_{I_R} & w_{m+1} & y_J & 1 \\
\hline
I \left\{ \begin{array}{c} \\ \\ \end{array} \right. & \begin{array}{c} y_{I_R} \\ y_{m+1} \end{array} & \begin{array}{c} M_{I_R I_R} \\ M^T_{I_R, m+1} \end{array} & \begin{array}{c} M_{I_R, m+1} \\ M_{m+1, m+1} \end{array} & \begin{array}{c} e \\ -1 \end{array} & \begin{array}{c} M_{I_R, m+1} \\ M_{m+1, m+1} \end{array} \\
\hline
J \{ & w_J & -e^T & 1 & 0 & 1 \\
\hline
\end{array}$$

Proof: In this case $\#I = m$, $\#I_R = m-1$, and $\#J = 1$. Therefore, using the notation introduced previously, we must have $m = r+1$, and $\ell = m-r = 1$. Hence the vector e is just the scalar 1. The result follows immediately from Corollary 3.3.1. ∎

Corollary 3.3.3. In the above situation (i.e. when tableau T^k has the property that $\#J = 1$), if the current solution $X = Pw$ is not optimal, then the next pivot to be executed by the S-algorithm must be either an in-pivot or an exchange-pivot.

Proof: Suppose on the contrary that an out-pivot were to be performed. In the resulting tableau, it will hold that $\#J = 0$, and hence $w_1 = w_2 = \dots = w_m = 0$. However, we also have $w_{m+1} = \sum_{i=1}^{m} w_i - 1 = 0$, a contradiction. ∎

Corollary 3.3.4. Starting from a tableau having $\#J > 1$, at most $\#J - 1$ consecutive out-pivots can be executed by the S-algorithm.

Proof: Follows from Corollary 3.4.3 and the fact that each out-pivot decreases $\#J$ by one. ∎

The following observation on positive semidefinite matrices is found in a paper by Cottle [1]:

Lemma 3.4. If M is a symmetric and positive semidefinite matrix and if $m_{kk} = 0$ for some k, then $m_{ik} = m_{ki} = 0$ for all $i \neq k$.

The next theorem deals with the case in which the index set J has cardinality greater than 1.

Theorem 3.5. Let T^k be a tableau generated by the S-algorithm in which $\#J \geq 2$. (Refer to the diagram in the statement of Corollary 3.3.1.) Then the symmetric positive semidefinite submatrix M_{JJ} has positive elements on its main diagonal.

Proof: Recall that M_{JJ} is a matrix of order ℓ. By Corollary 3.2.1(5), M_{JJ} is symmetric and positive semidefinite. By part (3) of the same Corollary, rank M_{JJ} = order $M_{JJ} - 1 = \ell-1$. Now suppose that more than one of the diagonal elements of M_{JJ} is 0, say $m_{11} = m_{22} = 0$. Using Lemma 3.4, we have:

$$M_{JJ} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & m_{33} & \cdots & m_{3\ell} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & m_{3\ell} & \cdots & m_{\ell\ell} \end{pmatrix}$$

This matrix must have rank less than or equal to $\ell-2$, which contradicts the fact that rank $M_{JJ} = \ell-1$.

The only remaining possibility is if one of the diagonal elements of M_{JJ} is zero, say m_{11}. Then

$$M_{JJ} = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & m_{22} & \cdots & m_{2\ell} \\ \vdots & \vdots & & \vdots \\ 0 & m_{2\ell} & \cdots & m_{\ell\ell} \end{pmatrix}$$

By hypothesis $m_{22}, m_{23}, \ldots, m_{\ell\ell}$ are all positive. Also, rank $M_{JJ} = \ell-1$. Hence the submatrix

$$\hat{m} = \begin{pmatrix} m_{22} & \cdots & m_{2\ell} \\ \vdots & & \vdots \\ m_{2\ell} & \cdots & m_{\ell\ell} \end{pmatrix}$$

has rank ℓ-1 and is thus nonsingular. But by Corollary 3.3.1, $e^T M_{JJ} = 0^T$. Therefore $\bar{e}^T \hat{M} = 0^T$ where $\bar{e} = (1,1,\ldots,1)^T$. But this contradicts the nonsingularity of $\hat{M}$. We conclude that $m_{11} > 0$, i.e. all of the main diagonal elements of M_{JJ} are positive. ∎

We now state the main result of this section.

Theorem 3.6. After the initialization step, all successive pivots performed by the S-algorithm are 1×1 principal pivots. Thus no "exchange-pivots" are necessary when the S-algorithm is applied to the LDP.

Proof: Let w_t be the driving variable in the current tableau. Either y_t (the negative basic distinguished variable which is the complement of w_t) blocks the increase of w_t by increasing to zero, or else some w_s $(s \in J)$ blocks the increase of w_t by decreasing to zero.

Case A: If y_t is the blocking variable, then clearly $m_{tt} = \partial y_t / \partial w_t > 0$. Hence, the in-pivot $\langle y_t, w_t \rangle$ may be made.

Case B: If w_s $(s \in J)$ is the blocking variable, and $\#J \geq 2$, Theorem 3.5 guarantees that $m_{ss} > 0$. This permits the out-pivot $\langle w_s, y_s \rangle$ to be made.

Consider finally the special case where $\#J = 1$. The tableau will then have the structure described in Corollary 3.3.2, and it will correspond to the primal feasible point P_s. Since $m_{ss} = 0$, an out-pivot is clearly impossible here. Furthermore, an exchange-pivot $\langle w_s, w_t \rangle$, $\langle y_t, y_s \rangle$ is also impossible, for it would yield a new tableau corresponding to the primal

feasible point P_t, which has a <u>higher</u> objective function value than P_s. (The fact that P_s is a point of minimum norm in P is guaranteed by the initialization step.) We conclude that if $\#J = 1$, the next pivot to be executed must be an in-pivot, and that y_t must be the blocking variable. ∎

<u>Remarks</u>

1. We have shown that, apart from the initialization step, no exchange-pivots in the S-algorithm are necessary. However, by Remark 3 following Proposition 3.1, the initialization can in fact be accomplished by two 1×1 principal pivots (rather than by an exchange-pivot). Thus, the S-algorithm can find the solution of the LDP by using 1×1 principal pivots exclusively.

2. Zoutendijk ([13], pp. 80-86) examined principal pivoting methods for solving linear complementarity problems of the form:

$$\begin{array}{c|c||c|} & y & 1 \\ \hline x & P^T P & -P^T d \\ \hline \end{array} \quad , \quad x^T y = 0,\ x \geq 0,\ y \geq 0.$$

These relations constitute the Kuhn-Tucker conditions for the problem:

$$\text{minimize } \frac{1}{2} \|d - Py\|^2$$

$$\text{subject to } \quad y \geq 0 \ .$$

Geometrically, this problem may be stated: Given a set of points $P = \{P_1, P_2, \ldots, P_m\}$ in E^n and a point $d \in E^n$, find the point closest to d in the convex cone generated by P. In the context of this problem, Zoutendijk proved results which are roughly analogous to Theorems 3.5 and 3.6 of this paper. He also suggests a number of different algorithmic procedures to resolve degeneracy.

3. An algorithm for finding the point of a finite cone (with given generators) nearest to a given point has been published by Wilhelmsen [10]. In a footnote, Wilhelmsen remarks that the algorithm can easily be modified to apply to convex polytopes and then it becomes identical to that of Wolfe [11,12].

c. Geometric interpretation of the tableaux elements

The following propositions provide a geometric interpretation of the elements in the successive tableaux generated by the S-algorithm. They are reformulations of Propositions 1-4 of Wolfe [11].

Assume that the current tableau T^k is given by:

	w_{I_R}	w_{m+1}	u_J	1
y_{I_R}	$M_{I_R I_R}$	$M_{I_R,m+1}$	$-M^T_{J,I_R}$	$M_{I_R,m+1}$
y_{m+1}	M_{m+1,I_R}	$M_{m+1,m+1}$	$-M^T_{J,m+1}$	$M_{m+1,m+1}$
w_J	M_{J,I_R}	$M_{J,m+1}$	M_{JJ}	$M_{J,m+1}$

As before, $I = I_R \cup \{m+1\}$ is the current set of indices of the basic dual variables, and J is the current set of indices of the basic primal variables.

Proposition 3.7. The elements of the vector $M_{J,m+1}$ are the coordinates of the current feasible point X in the set $Q = \{P_j : j \in J\}$.

Proof: We have $X = Pw$ where

$$w = \begin{bmatrix} w_{I_R} \\ w_J \end{bmatrix}$$

In the tableau T^k corresponding to X, however, w_{I_R} is nonbasic. Also, by the duplicate column property of the tableau, the value of the basic variables w_J is given by $M_{J,m+1}$. Hence

$$w = \begin{bmatrix} w_{I_R} \\ w_J \end{bmatrix} = \begin{bmatrix} 0 \\ M_{J,m+1} \end{bmatrix} \geq 0$$

by feasibility. By Corollary 3.3.1,

$$\sum_{i=1}^{m} w_i = e^T M_{J,m+1} = 1 .$$

Remark. After each out-pivot is performed, we obtain a "non-complementary" tableau in the sense that $w_t y_t > 0$ for some $t \in I$. In such a tableau, the values of all primal basic variables w_j $(j \in J)$ must be modified to be $q_j + m_{jt}\delta$, where δ is the value at which the driving variable w_t (now nonbasic) was blocked in the previous tableau, and $q_j = M_{j,m+1}$. Furthermore, the nonbasic driving variable w_t is assigned a positive value δ in the "non-complementary" tableau.

Proposition 3.8.

(1) In a complementary tableau, the element $M_{m+1,m+1}$ gives the current value of y_{m+1}, which is equal to $w^T P^T P w = X^T X$ (where $X = Pw$ is the current primal feasible point).

(2) In a non-complementary tableau, the following identity holds:

$$X^T X = w^T P^T P w = M_{m+1,m+1} + 2M_{t,m+1} + M_{tt}\delta^2$$

where δ is the value at which the increase of the driving variable w_t, $t \in I_R$, was blocked; $M_{t,m+1}$ is the component of the vector $M_{I_R,m+1}$ corresponding to the variable y_t; and M_{tt} is the corresponding diagonal entry of the matrix $M_{I_R I_R}$.

Proof: These results may be readily demonstrated using the approach of van de Panne and Whinston [8], who adjoin a row representing the objective function to the Kuhn-Tucker tableau and then perform a principal transformation on the augmented tableau. Alternatively, the proposition may be proven directly, using Corollary 3.3.1. ∎

Proposition 3.9. In a complementary tableau, the value of the element $M_{i,m+1}$ $(i \in I_R)$ is $P_i^T X - X^T X$, where $X = Pw$ is the current primal feasible point.

Proof: Consider the initial tableau T^0 shown in the proof of Corollary 3.3.1. Perform a block pivot on the submatrix

$$\begin{bmatrix} Q^T Q & -e \\ e^T & 0 \end{bmatrix} .$$

A straightforward calculation shows that in tableau T^k,

$$M_{I_R,m+1} = R^T Q M_{J,m+1} - \hat{e}\, M_{m+1,m+1} \cdot$$

Hence the ith component of this vector is just

$$\begin{aligned} M_{i,m+1} &= P_i^T Q M_{J,m+1} - M_{m+1,m+1} \\ &= P_i^T X - X^T X \end{aligned}$$

by Propositions 3.7 and 3.8. ∎

Corollary 3.9.1. In a complementary tableau, the absolute value of the element $M_{i,m+1}$ is $\|X\|$ times the distance of P_i from the hyperplane $H(X) = \{Y \in E^n : X^T Y = X^T X\}$. If P_i lies on the near side of $H(X)$, the element $M_{i,m+1}$ will be a negative number; hence in this case we take the sign of the distance to be negative.

Proof: Let Y denote the projection of P_i onto $H(X)$, i.e. the point of $H(X)$ nearest to P_i. Since X is normal to $H(X)$, $Y-P_i = \mu X$ for some scalar μ. Premultiplying by $X^T \neq 0$ yields $X^T Y - X^T P_i = \mu X^T X$. But since $Y \in H(X)$, $X^T Y = X^T X$. Hence we have:

$$\mu = \frac{X^T X - X^T P_i}{X^T X} \quad .$$

Now the distance of P_i from $H(X)$ is:

$$\|Y - P_i\| = \|\mu X\| = |\mu|\ \|X\|$$

Hence

$$\|X\| \cdot \|Y - P_i\| = |\mu| \|X\|^2 = |\mu|\ X^T X = |X^T X - X^T P_i| = |M_{i,m+1}|$$

by Proposition 3.9.

Finally, observe that if P_i lies on the near side of $H(X)$ then $P_i^T X < X^T X$. In such a case $M_{i,m+1} = P_i^T X - X^T X < 0$, and so $M_{i,m+1} = -\|X\| \cdot \|Y - P_i\|$. ∎

Proposition 3.10. Let $Q = \{P_j : j \in J\}$. Define the point W_i to be the projection of the point P_i onto the affine hull $A(Q) = \{Z : Z = Qu,\ e^T u = 1\}$, where $i \in I_R = I \setminus \{m+1\}$. Then:

(1) For $i \in I_R$, the ith row of the matrix $-M^T_{J,I_R}$ gives the coordinates u of the point W_i with respect to the set of points Q. In other words,

$$[-M^T_{J,I_R}]_{i.} = u^T \quad \text{and} \quad W_i = Qu \ .$$

(2) The entry M_{ii} $(i \in I_R)$ on the main diagonal of $M_{I_R I_R}$ equals the square of the distance of P_i from $A(Q)$.

Proof:

(1) The problem of projecting P_i onto $A(Q)$ may be formulated as: minimize $\|P_i - Qu\|^2$, subject to $e^T u = 1$. As shown by Wolfe [10], necessary and sufficient conditions for a solution to this problem are:

$$\begin{aligned} Q^T Qu - e\lambda &= Q^T P_i \\ e^T u &= 1 \end{aligned}$$

Employing the notation of Lemma 3.3 and Corollary 3.3.1, we have

$$\begin{bmatrix} u \\ \lambda \end{bmatrix} = \begin{bmatrix} Q^T Q & -e \\ e^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} Q^T P_i \\ 1 \end{bmatrix} = \begin{bmatrix} \bar{M}_{22} & \bar{M}_{23} \\ \bar{M}_{32} & \bar{M}_{33} \end{bmatrix} \begin{bmatrix} Q^T P_i \\ 1 \end{bmatrix} \ .$$

Hence $u = \bar{M}_{22}Q^TP_i + \bar{M}_{23} = M_{JJ}Q^TP_i + M_{J,m+1}$. However, it can also be shown that

$$-M^T_{J,I_R} = R^TQ\bar{M}_{22} + e\bar{M}_{23} = R^TQM_{JJ} + eM^T_{J,m+1}$$

The ith row of this matrix is clearly identical to u^T. Now, setting $W_i = Qu$ yields the projection of P_i onto $A(Q)$.

(2) Using Lemma 3.3 and Corollary 3.3.1, we obtain:

$$M_{I_R,I_R} = R^TR - R^TQM_{JJ}Q^TR - R^TQM_{J,m+1}e^T - \hat{e}M^T_{J,m+1}Q^TR + \hat{e}M_{m+1,m+1}e^T.$$

The (i,i) diagonal element of this matrix reduces to

$$M_{ii} = P_i^TP_i - P_i^TQu - P_i^TQu + W_i^TW_i = (P_i - W_i)^T(P_i - W_i) = \|P_i - W_i\|^2.$$

▌

4. Equivalence of the algorithms

To show that the S-algorithm and Wolfe's LDP algorithm generate the same feasible points, we need to demonstrate that the corresponding steps of the two methods achieve identical results.

Assume that, at some iteration of the S-algorithm, a basic complementary tableau T^k has been obtained, corresponding to a primal feasible (but not optimal) point X^k. Such a point is denoted by Wolfe as a point of minimum norm in the convex hull $C(Q^k)$ of a certain corral Q^k.

Suppose that the tableau has the following structure:

$$
\begin{array}{lll|ccc|cc||c}
 & & & w_t & w_{\bar{I}} & w_{m+1} & y_s & y_{\bar{J}} & 1 \\
\hline
\multirow{3}{*}{I} & \multirow{2}{*}{I_R} & y_t & M_{tt} & M^T_{\bar{I}t} & M_{t,m+1} & -M_{st} & -M^T_{\bar{J}t} & M_{t,m+1} \\
 & & y_{\bar{I}} & M_{\bar{I}t} & M_{\bar{I}\bar{I}} & M_{\bar{I},m+1} & -M^T_{s\bar{I}} & -M^T_{\bar{J}\bar{I}} & M_{\bar{I},m+1} \\
 & & y_{m+1} & M_{t,m+1} & M^T_{\bar{I},m+1} & M_{m+1,m+1} & -M_{s,m+1} & -M^T_{\bar{J},m+1} & M_{m+1,m+1} \\
\hline
\multirow{2}{*}{J} & & w_s & M_{st} & M_{s\bar{I}} & M_{s,m+1} & M_{ss} & M^T_{\bar{J}s} & M_{s,m+1} \\
 & & w_{\bar{J}} & M_{\bar{J}t} & M_{\bar{J}\bar{I}} & M_{\bar{J},m+1} & M_{\bar{J}s} & M_{\bar{J}\bar{J}} & M_{\bar{J},m+1} \\
\end{array}
$$

where $I = \{t\} \cup \bar{I} \cup \{m+1\} = I_R \cup \{m+1\}$, and $J = \{s\} \cup \bar{J}$. Let q denote the constant column.

The feasible point X^k corresponding to this tableau has barycentric coordinates $w_{I_R} = 0$ and $w_J > 0$ with respect to the set of points P. In Wolfe's terminology, the current corral Q^k is composed of the points P_s and $\{P_j : j \in \bar{J}\}$.

Without loss of generality, we may assume that the most negative basic dual variable is y_t, which has the value $M_{t,m+1} < 0$. Hence the S-algorithm does not recognize the current point X^k as being optimal. By Proposition 3.9, $M_{t,m+1} = P_t^T X^k - X^{k^T} X^k$, so $P_t^T X^k < X^{k^T} X^k$. By Theorem 2.1 of Wolfe [11], X is the solution of the LDP if and only if $X^T P_j \geq X^T X$; step 1 of Wolfe's method determines whether this condition is satisfied. Thus, in this case the current feasible point X^k would be found non-optimal by Wolfe's method.

In this situation, an attempt is made in the S-algorithm to increase w_t (the complement of the distinguished variable y_t). By Lemma 3.4, it follows that $M_{tt} > 0$ (for if $M_{tt} = 0$, then $M_{t,m+1} = 0$, contradicting the assumption that $M_{t,m+1} < 0$).

A preliminary ratio test is then performed to find an index s such that

$$\frac{q_s}{M_{st}} = \max_{j \in J} \left\{ \frac{q_j}{M_{jt}} : M_{jt} < 0 \right\} .$$

In the notation of the above tableau, we have

$$\frac{M_{s,m+1}}{M_{st}} \geq \frac{M_{j,m+1}}{M_{jt}} \qquad \text{for all } j \in \bar{J} \text{ for which } M_{jt} < 0.$$

Since $M_{s,m+1} = w_s > 0$ by the nondegeneracy assumption,* it follows that

$$\frac{M_{s,m+1}}{M_{st}} < 0 .$$

The next step of the S-algorithm is to determine whether the basic variable first blocking the increase of w_t (i.e. attaining the value of zero) is y_t or w_s. To accomplish this, two ratios are compared, namely

$$\frac{M_{t,m+1}}{M_{tt}} < 0 \qquad \text{and} \qquad \frac{M_{s,m+1}}{M_{st}} < 0 .$$

---

* See Appendix I.

There are two cases to consider:

Case A. Suppose that

$$\frac{M_{t,m+1}}{M_{tt}} \geq \frac{M_{s,m+1}}{M_{st}} .$$

Then y_t blocks the increase of w_t before (or simultaneously as) w_s decreases to 0. (The S-algorithm always resolves blocking ties in favor of the distinguished variable y_t.) In this situation, the in-pivot $\langle y_t, w_t \rangle$ is executed to yield a new tableau T^{k+1} corresponding to a point X^{k+1} which is primal feasible. As will be shown, that is precisely the same pivot as is performed in Step 2 of the tableau variant of Wolfe's LDP algorithm.

In the tableau T^{k+1} resulting from the in-pivot, the basic primal variables are w_t, w_s and $w_{\bar{J}}$, and their values are given by:

$$\begin{array}{c|l|}
\cline{2-2}
w_t & \quad - \dfrac{M_{t,m+1}}{M_{tt}} \\[2ex]
w_s & M_{s,m+1} - \dfrac{M_{t,m+1} \cdot M_{st}}{M_{tt}} \\[2ex]
w_{\bar{J}} & M_{\bar{J},m+1} - \dfrac{M_{t,m+1} \cdot M_{\bar{J}t}}{M_{tt}} \\
\cline{2-2}
\end{array}$$

In Wolfe's terminology, the new corral Q^{k+1} is composed of the points P_t, P_s, and $\{P_j : j \in \bar{J}\}$. The new feasible point X^{k+1} has barycentric coordinates w_t, w_s, and $w_{\bar{J}}$ with respect to the set of points Q^{k+1}.

To demonstrate the fact that the two algorithms generate the same point, it must be shown that X^{k+1} is identical to Y, the point of smallest norm in $A(Q^{k+1})$, and furthermore that $X \in \text{rel int } C(Q^{k+1})$.

Since the ratio test indicated that an in-pivot was possible, the point $X^{k+1} = Pw$ must be primal feasible. Hence $w \geq 0$ and $e^T w = 1$; in particular, $w_t + w_s + \sum_{j \in \bar{J}} w_j = 1$. Assuming nondegeneracy, all of the barycentric coordinates of X^{k+1} are positive. Hence $X^{k+1} \in \text{rel int } C(Q^{k+1})$.

Now, let Y be the point of smallest norm in $A(Q^{k+1})$. Y can be determined by solving the problem:

$$\text{minimize } \|Qu\|^2 \quad \text{subject to} \quad e^T u = 1, \quad \text{where} \quad Q \equiv Q^{k+1}.$$

Substituting the point 0 for the point P_i in the proof of Proposition 3.10, we find that the barycentric coordinates of Y with respect to the set Q^{k+1} are given by:

$$u = M_{JJ} Q^T P_i + M_{J,m+1} = M_{J,m+1} .$$

But the vector $M_{J,m+1}$ represents the current values of the basic variables

$$\begin{bmatrix} w_t \\ w_s \\ w_{\bar{J}} \end{bmatrix}$$

in Tableau T^{k+1}, since $J = \{t\} \cup \{s\} \cup \bar{J}$. In other words, $M_{J,m+1}$ is the vector of barycentric coordinates of the point X^{k+1} with respect to the set of points Q^{k+1}.

Since Q^{k+1} is an affinely independent set, any point in $A(Q^{k+1})$ will have a unique barycentric representation with respect to Q^{k+1}. Because X^{k+1} and Y have the <u>same</u> barycentric coordinates, namely $M_{J,m+1} = u$, they must be identical. Hence the point X^{k+1} obtained after an in-pivot of the S-algorithm is indeed the point of minimum norm in $A(Q^{k+1})$, and furthermore, $X^{k+1} \in \text{rel int } C(Q^{k+1})$.

<u>Case B.</u> Suppose the ratio test indicates that

$$\frac{M_{t,m+1}}{M_{tt}} < \frac{M_{s,m+1}}{M_{st}}$$

Then, w_s blocks the increase of w_t (i.e. decreases to 0) before y_t increases to 0. In this case, the S-algorithm performs the <u>out-pivot</u> $\langle w_s, y_s \rangle$ in the tableau T^k. As a result, a new tableau is obtained which we denote as T^* (to avoid confusion with T^{k+1} considered in Case A). For the purpose of the proof. the relevant portion of the tableau T^* is given by:

	$w_t = \delta$	1
$w_{\bar{J}}$	$M_{\bar{J}t} - \frac{M_{st}}{M_{ss}} M_{\bar{J}s}$	$M_{\bar{J},m+1} - \frac{M_{s,m+1}}{M_{ss}} M_{\bar{J}s}$

Tableau T^* is "noncomplementary" in the sense that the driving variable w_t, although nonbasic, has a positive value δ: the value at which its increase was blocked by w_s. The value of δ is determined by the ratio test to be:

$$\delta = -\left(\frac{M_{s,m+1}}{M_{st}}\right) .$$

Since w_t has the value δ in tableau T^*, the values of the basic primal variables $w_{\bar{J}}$ must be adjusted accordingly:

$$w_{\bar{J}} = \left[M_{\bar{J}t} - \frac{M_{st}}{M_{ss}} M_{\bar{J}s}\right]\delta + M_{\bar{J},m+1} - \frac{M_{s,m+1}}{M_{ss}} M_{\bar{J}s}$$

$$= -\frac{M_{s,m+1}}{M_{st}} M_{\bar{J}t} + M_{\bar{J},m+1} .$$

Let $Q^* = \{P_t\} \cup \{P_j : j \in \bar{J}\}$, and let X^* be the primal feasible point corresponding to the new tableau T^*. Then the barycentric coordinates of X^* with respect to the set Q^* are given by

$$\begin{bmatrix} w_t \\ w_{\bar{J}} \end{bmatrix}$$

where w_t has the value $\delta > 0$ specified above. Since X^* is feasible, we must have

$$\begin{bmatrix} w_t \\ w_{\bar{J}} \end{bmatrix} \geq 0$$

and $w_t + \sum_{j\in\bar{J}} w_j = 1$.

To summarize the notation used here, we have:

$Q^k = \{P_s\} \cup \{P_j : j \in \bar{J}\}$ = original corral corresponding to tableau T^k

X^k = original feasible point corresponding to T^k

$Q^{k+1} = \{P_t\} \cup \{P_s\} \cup \{P_j : j \in \bar{J}\}$

= corral obtained by adjoining the point P_t to Q^{k+1} by means of an in-pivot in T^k.

Y = point of smallest norm in $A(Q^{k+1})$. As shown in case A, this point would have been generated by an in-pivot in T^k had the ratio test permitted such a pivot.

$Q^* = \{P_t\} \cup \{P_j : j \in \bar{J}\}$

X^* = point generated by an out-pivot $\langle w_s, y_s \rangle$ in T^k.

Finally, define Z to be the point nearest to Y on the line segment $C(Q^{k+1}) \cap \overline{X^kY}$. The point Z is calculated in Step 3 of Wolfe's LDP method. To demonstrate that the S-algorithm arrives at the same point, it suffices to show that $Z = X^*$. Thus, there are three facts to be proved:

(1) $X^* \in C(Q^{k+1})$

(2) $X^* \in \overline{X^kY}$

(3) Any movement from X^* toward Y along the line segment $\overline{X^kY}$ will result in a point which is exterior to Q^{k+1}.

To prove that X^* can be expressed as a convex combination of X^k and Y, we must show that $0 < \lambda < 1$ in the equation:

$$X^* = (1-\lambda)X^k + \lambda Y .$$

This equation can be rewritten in terms of the barycentric coordinates of the points X^*, X^k, and Y with respect to the entire set $P = \{P_{\bar{I}}, P_t, P_s, P_{\bar{J}}\}$. Using facts established earlier in this discussion we have:

$$\begin{bmatrix} w_{\bar{I}} \\ w_t \\ w_s \\ w_{\bar{J}} \end{bmatrix} = \begin{bmatrix} 0 \\ -\dfrac{M_{s,m+1}}{M_{st}} \\ 0 \\ -\dfrac{M_{s,m+1}}{M_{st}} M_{\bar{J}t} + M_{\bar{J},m+1} \end{bmatrix}$$

$$= (1-\lambda) \begin{bmatrix} 0 \\ 0 \\ M_{s,m+1} \\ M_{\bar{J},m+1} \end{bmatrix} + \lambda \begin{bmatrix} 0 \\ -\dfrac{M_{t,m+1}}{M_{tt}} \\ M_{s,m+1} - \dfrac{M_{t,m+1} \cdot M_{st}}{M_{tt}} \\ M_{\bar{J},m+1} - \dfrac{M_{t,m+1} \cdot M_{\bar{J}t}}{M_{tt}} \end{bmatrix}$$

Solving this vector equation for λ, we obtain

$$\bar{\lambda} = \frac{M_{tt}}{M_{t,m+1}} \cdot \frac{M_{s,m+1}}{M_{st}} .$$

$\bar{\lambda}$ is positive since $M_{tt} > 0$, $M_{t,m+1} < 0$, $M_{s,m+1} > 0$ and $M_{st} < 0$ by hypothesis. Furthermore, the outcome of the ratio test indicates that

$$\frac{M_{t,m+1}}{M_{tt}} < \frac{M_{s,m+1}}{M_{st}},$$

hence $M_{t,m+1}M_{st} > M_{tt}M_{s,m+1}$, which implies that $\bar{\lambda} < 1$. We conclude that $X^{*} = (1-\bar{\lambda})X^{*} + \bar{\lambda}Y$ with $0 < \bar{\lambda} < 1$, i.e. X^{*} can be expressed as a convex combination of X^{*} and Y. Also, since $X^{*} \in C(Q^{*})$ and $Q^{*} \subset Q^{k+1}$, it follows that $X^{*} \in C(Q^{k+1})$.

It remains to verify fact (3). In the vector equation above, note that the value of the w_s-coordinate as a function of λ is given by:

$$w_s = (1-\lambda)M_{s,m+1} + \lambda M_{s,m+1} - \lambda \frac{M_{t,m+1} \cdot M_{st}}{M_{tt}} = M_{s,m+1} - \lambda \frac{M_{t,m+1} \cdot M_{st}}{M_{tt}}$$

If $\lambda = \bar{\lambda}$, we have $w_s = 0$. For feasibility (i.e. $X^{*} \in C(Q^{k+1})$), we require that $w_s \geq 0$. Setting

$$M_{s,m+1} - \lambda \frac{M_{t,m+1} \cdot M_{st}}{M_{tt}} \geq 0$$

implies that $\lambda \leq \bar{\lambda}$. In other words, any increase in λ beyond the value $\bar{\lambda}$ will result in an infeasible point.

To complete the equivalence proof, it must be shown that the critical value $\bar{\lambda}$ is in fact identical to the value $\bar{\theta}$ determined in Step 3 of Wolfe's LDP algorithm. This step calculates the ratio:

$$\bar{\theta} = \min \left\{ \frac{w_j}{w_j - y_j} : w_j - y_j > 0 \right\}$$

where w denotes the barycentric coordinates of X^k with respect to P, and y denotes the barycentric coordinates of Y with respect to P. But these coordinates are just:

$$w = \begin{bmatrix} 0 \\ 0 \\ M_{s,m+1} \\ M_{\bar{J},m+1} \end{bmatrix} \quad \text{and} \quad y = \begin{bmatrix} 0 \\ -\dfrac{M_{t,m+1}}{M_{tt}} \\ M_{s,m+1} - \dfrac{M_{t,m+1} \cdot M_{st}}{M_{tt}} \\ M_{\bar{J},m+1} - \dfrac{M_{t,m+1} \cdot M_{\bar{J}t}}{M_{tt}} \end{bmatrix}$$

Thus,

$$\bar{\theta} = \min\left\{\left[M_{s,m+1}\left(\frac{M_{t,m+1} \cdot M_{st}}{M_{tt}}\right)^{-1}\right], \min\left[M_{j,m+1}\left(\frac{M_{t,m+1} \cdot M_{jt}}{M_{tt}}\right)^{-1} : j \in \bar{J}\right]\right\}$$

By the ratio test,

$$\frac{M_{s,m+1}}{M_{st}} \geq \frac{M_{j,m+1}}{M_{jt}} \quad \text{for all } j \in \bar{J} \text{ for which } \bar{M}_{jt} < 0.$$

Since $M_{tt}/M_{t,m+1} < 0$, it follows that

$$M_{s,m+1}\left(\frac{M_{t,m+1} \cdot M_{st}}{M_{tt}}\right)^{-1} \leq M_{j,m+1}\left(\frac{M_{t,m+1}\ M_{jt}}{M_{tt}}\right)^{-1} \quad \text{for all such } j.$$

Hence

$$\bar{\theta} = M_{s,m+1}\left(\frac{M_{t,m+1} \cdot M_{st}}{M_{tt}}\right)^{-1}$$

which is identical to the critical value $\bar{\lambda}$ calculated earlier. We conclude that in Case B, the S-algorithm and Wolfe's LDP algorithm generate identical primal feasible points. ∎

Remark. The only difference between the two algorithms lies in the fact that the S-algorithm first performs a feasibility test (namely, the ratio test) to determine whether an in-pivot or an out-pivot should be executed. The former case leads to the start of a new major cycle; the latter case corresponds to a minor cycle. Note that all tableaux generated by the algorithm correspond to primal feasible points.

The tableau variant of Wolfe's LDP algorithm, on the other hand, first performs the in-pivot, then performs a test to determine whether the point Y corresponding to the new tableau is feasible (i.e., lies in rel int $C(Q^{k+1})$). If so, the algorithm begins a new major cycle. However, it is possible for the in-pivot to generate a tableau corresponding to a point Y which is infeasible. In this case, it is then necessary to enter a minor cycle (step (3)), which involves calculating the ratio $\bar{\theta}$ and setting the new point equal to $(1-\bar{\theta})X^k + \bar{\theta}Y$. As has been shown, this minor cycle achieves the same result as does the out-pivot step of the S-algorithm. Thus, although the two methods generate different tableaux in their respective minor cycles, the sequences of primal feasible points determined by the two algorithms are identical. This point is illustrated by applying the two algorithms to Wolfe's example:

Example.

Assume that we start from the feasible point R having barycentric coordinates (9/13, 4/13, 0). The corresponding tableau is:

$$\begin{array}{c c c} & & \begin{array}{cccc|c} w_3 & w_4 & y_1 & y_2 & 1 \end{array} \\ \begin{array}{c} y_3 \\ y_4 \\ w_1 \\ w_2 \end{array} & = \frac{1}{13} & \left[\begin{array}{cccc||c} 49 & -42 & 17 & -4 & -42 \\ -42 & 36 & -9 & -4 & 36 \\ \hline -17 & 9 & 1 & -1 & 9 \\ 4 & 4 & -1 & 1 & 4 \end{array}\right] \end{array}$$

S-algorithm	Wolfe's LDP algorithm

S-algorithm

Increase the driving variable w_3. Ratio test shows that w_1 blocks.

$\boxed{\text{OUT-PIVOT } \langle w_1, y_1 \rangle}$

$$\begin{array}{c c} & \begin{array}{cccc|c} w_1 & w_3{=}\delta & w_4 & y_2 & 1 \end{array} \\ \left[\begin{array}{c} y_1 \\ y_3 \\ y_4 \\ w_4 \end{array}\right] & \left[\begin{array}{cccc||c} 13 & 17 & -9 & 1 & -9 \\ 17 & 26 & -15 & 1 & -15 \\ -9 & -15 & 9 & -1 & 9 \\ \hline -1 & -1 & 1 & 0 & 1 \end{array}\right] \end{array}$$

w_3 has the value $\delta = 9/17$. Adjusting the values of the variables w this tableau

Wolfe's LDP algorithm

Without regard to feasibility, perform the pivot

$\boxed{\langle y_3, w_3 \rangle}$

$$\begin{array}{c c c} & & \begin{array}{cccc|c} w_4 & y_1 & y_2 & y_3 & 1 \end{array} \\ \left[\begin{array}{c} y_4 \\ w_1 \\ w_2 \\ w_3 \end{array}\right] & = \frac{1}{49} & \left[\begin{array}{cccc||c} 0 & 21 & -28 & -42 & 0 \\ \hline -21 & 26 & -9 & -17 & -21 \\ 28 & -9 & 5 & 4 & 28 \\ 42 & -17 & 4 & 13 & 42 \end{array}\right] \end{array}$$

This tableau corresponds to the infeasible point O with barycentric coordinates (-3/7, 4/7, 6/7).

corresponds to the point S with barycentric coordinates (0, 8/16, 9/17). Continue the increase of w_3. Ratio test shows that y_3 blocks.

IN-PIVOT $\langle y_3, w_3 \rangle$

$$
\begin{array}{c}
\\
\end{array}
\begin{bmatrix} y_1 \\ y_4 \\ w_2 \\ w_3 \end{bmatrix} = \frac{1}{26}
\begin{array}{ccccc}
w_1 & w_2 & y_2 & y_4 & 1 \\
\end{array}
$$

$$
\begin{bmatrix} y_1 \\ y_4 \\ w_2 \\ w_3 \end{bmatrix} = \frac{1}{26}
\left[\begin{array}{cccc|c}
49 & 21 & 9 & 17 & 21 \\
21 & 9 & -11 & -15 & 9 \\
\hline
-9 & 11 & 1 & -1 & 11 \\
-17 & 15 & 11 & 1 & 15
\end{array}\right]
$$

Calculate $\bar{\theta} = 21/34$. Computing $(1-\bar{\theta})R + \bar{\theta}Q$ yields the point S. Since $\bar{\theta}$ is attained for the index $i = 1$, perform the pivot

$\langle w_1, y_1 \rangle$

$$
\begin{array}{ccccc}
w_1 & w_4 & y_2 & y_4 & 1
\end{array}
$$

$$
\begin{bmatrix} y_1 \\ y_4 \\ w_2 \\ w_3 \end{bmatrix} = \frac{1}{26}
\left[\begin{array}{cccc|c}
49 & 21 & 9 & 17 & 21 \\
21 & 9 & -1 & -15 & 9 \\
\hline
-9 & 11 & 1 & -1 & 11 \\
-17 & 15 & -1 & 1 & 15
\end{array}\right]
$$

The preceding results may be summarized as follows:

Theorem 4.1.

(1) When applied to the LDP, the S-algorithm of van de Panne and Whinston and Wolfe's LDP algorithm are equivalent in the sense that they generate the same sequence of primal feasible points.

(2) Furthermore, the successive tableaux generated by the two methods are identical, except during certain iterations in which Wolfe's algorithm produces a tableau corresponding to an infeasible point. In such cases Wolfe's method performs a minor cycle to restore feasibility; this step achieves the same result as does the corresponding minor cycle in the S-algorithm.

(3) Both the S-algorithm and the tableau variant of Wolfe's method rely exclusively on 1×1 principal pivots in solving the LDP. The same principal pivots are executed in both methods, although possibly in different order.

Remarks.

1. van de Panne and Whinston [8] showed that in the case of convex quadratic programming, the S-algorithm generates the same sequence of primal feasible points as does the "asymmetric" algorithm due to Dantzig [3]. Since the LDP is a special case of the convex quadratic programming problem, Wolfe's LDP algorithm is also equivalent (in the sense described above) to Dantzig's method applied to such a problem.

2. From the computational point of view, the S-algorithm and the tableau variant of Wolfe's algorithm are also equivalent. The "duplicate column" property and the bisymmetry of the tableaux generated by the S-method imply that $(m+1)^2/2$ storage locations are required.

It might appear that performing a minor cycle (step (3)) in Wolfe's method (which involves calculating a parameter $\bar{\theta}$ and determining the new feasible point as a convex combination of X^k and Y) entails additional computational effort beyond that required in the S-algorithm. This is not the case, however, since the latter algorithm performs the equivalent computation of the ratio test and the adjustment in the values of the basic variables after an out-pivot has taken place.

Appendix I: On degeneracy

In order to guarantee that the S-algorithm will solve the LDP in a finite number of pivot steps, it is not necessary to make the nondegeneracy assumption (i.e. the assumption that in every tableau, the basic primal variables are all strictly positive).

To see this, suppose that the current tableau T^k has the structure depicted in the first diagram of Section 4. Assume that the current complementary solution is not optimal, and that y_t is the most negative basic dual variable (so that $M_{t,m+1} < 0$). Now, contrary to the nondegeneracy assumption, suppose that $M_{s,m+1} = 0$. (The other basic primal variables comprising the vector $M_{\bar{J},m+1}$ may be taken to be positive.)

The ratio test of the S-algorithm will determine that

$$\frac{M_{t,m+1}}{M_{tt}} < \frac{M_{s,m+1}}{M_{st}} = 0 \ .$$

Hence "Case B" obtains and an out-pivot must be executed on the element M_{ss}. (M_{ss} is positive by Theorem 3.5.) The resulting tableau T^*, as indicated in the discussion of the preceding section, will have:

$$w_{\bar{J}} = M_{\bar{J},m+1} - \frac{M_{s,m+1}}{M_{ss}} M_{\bar{J}s} = M_{\bar{J},m+1} - 0 > 0 \ .$$

Hence, a zero-valued basic barycentric coordinate (basic primal variable) in any tableau may be "removed" by making an out-pivot.

If several basic barycentric coordinates in a given tableau have the value zero, it will be necessary to perform a sequence of out-pivots. Corollary 3.3.3 sets an upper bound on the number of such pivots that can be executed. Eventually, either of two cases must arise:

(1) The cardinality of J is 1 and the tableau has the nondegenerate structure described in Corollary 3.3.1, or

(2) The cardinality of J is greater than 1, but the ratio test indicates that an in-pivot should be performed. This, of course, happens only if all of the basic barycentric coordinates are positive, i.e. the tableau is nondegenerate.

Appendix II: von Hohenbalken's method

In this section, an algorithm due to von Hohenbalken [9] for maximizing certain pseudoconcave functions on polytopes is considered. It will be shown that this algorithm, when specialized to the LDP, is identical to Wolfe's method.

In the following description of von Hohenbalken's method, the notation employed in section 2 of this paper will be used. In addition, certain maximization problems will be converted to equivalent minimization problems for the sake of clarity.

Following the initial step, the algorithm continues with a sequence of major cycles, each of which begins at Basic Step 1. Let Q^k be the affinely independent subset of P at major cycle k, and let X^k be the minimizer of $f(X)$ on $C(Q^k)$, where X^k belongs to the relative interior of $C(Q^k)$. (Thus, Q^k is a "corral" in Wolfe's terminology.)

VON HOHENBALKEN'S ALGORITHM	SPECIALIZATION TO THE LDP
<u>Initial Step</u> Find an extreme point of the feasible region. In general, this is accomplished by solving a linear program. Denote this extreme point as $X^{k+1} = X^1$ (the superscript refers to the cycle number). Go to basic step 1.	In the case of the LDP, without loss of generality we may take this extreme point to be one having minimum norm; it may be denoted by $P_{\hat{i}}$. With this choice, the same starting point is selected as in Wolfe's LDP method. Set $X^1 = P_{\hat{i}}$. Go to basic step 1.
<u>Basic Step 1</u>. Set $X^k = X^{k+1}$. Use linear programming to determine an extreme point $\hat{X}$ of the feasible region that solves: $\min\{\nabla f(X^k)^T X : X \text{ is feasible}\}$ Optimality test: (a) If $\nabla f(X^k)^T[\hat{X}-X^k] = 0$, stop: X^k is optimal. (b) If $\nabla f(X^k)^T[\hat{X}-X^k] < 0$, go to Basic Step 2.	In the case of the LDP, it is unnecessary to use linear programming to solve the minimization problem of this step, since the problem is just: $\min_{X \in C(P)} 2X^{k^T} X$ which is equivalent to: $\min_{P_i \in P} X^{k^T} P_i$. This computation is equivalent to determining an extreme point P_j as that point on the near side of the hyperplane $H(X^k) = \{Z : X^{k^T} Z = X^{k^T} X^k\}$ having greatest distance from the hyperplane. But this is precisely

VON HOHENBALKEN'S ALGORITHM	SPECIALIZATION TO THE LDP
	the step taken in Step 1 of Wolfe's method (see Wolfe [11, pages 10 and 23]). In the case of the LDP, $\hat{X}$ is just the extreme point P_j which minimizes $X^{k^T} P_i$. Hence the expression in Basic Step 1(a) is equivalent to $2X^{k^T} P_j - 2X^{k^T} X^k = 0$. The algorithm terminates, therefore, if $X^{k^T} X^k = X^{k^T} P_j$. In this case X^k is the optimal solution of the LDP. Note that Wolfe's method employs the same optimality test: Theorem 2-1 (Wolfe [11]) states that X is optimal for the LDP if $X^T P_j \geq X^T X$ for all j. If $\nabla f(X^k)^T [\hat{X} - X^k] < 0$ (i.e. if $X^{k^T} P < X^{k^T} X^k$), then X^k is clearly not optimal and the method continues.

VON HOHENBALKEN'S ALGORITHM

Basic Step 2

There are two cases:

(a) If $f(\hat{X}) < f(X^k)$, set

$$X^{k+1} = \hat{X},$$

$$Q^{k+1} = \{\hat{X}\} ,$$

and go to basic step 1.

(b) If $f(\hat{X}) \geq f(X^k)$, augument Q^k by $\hat{X}$ to form a new affinely independent subset Q. Go to basic step 3.

SPECIALIZATION TO THE LDP

When the method is applied to the LDP, case (a) will never occur. Recall that the starting point $P_{\hat{i}}$ was chosen to have minimum norm, i.e. $P_{\hat{i}}^T P_{\hat{i}} \leq P_i^T P_i$ for all i. Hence, $\hat{X} = P_j$ and $X^1 = P_{\hat{i}}$, we must have:

$$f(\hat{X}) = P_j^T P_j \geq P_{\hat{i}}^T P_{\hat{i}} = X^{1T} X^1 = f(X^1)$$

So case (b) clearly occurs here. Now, let X^k be the feasible point available at the beginning of major cycle k. It is clear that

$$f(X^k) = X^{k^T} X^k \leq P_{\hat{i}}^T P_{\hat{i}} \leq P_i^T P_i, \text{ all } i$$

Since $\hat{X}$ must be one of the points P_i, it follows that $f(\hat{X}) \geq f(X^k)$, i.e. case (b) occurs here as well.

VON HOHENBALKEN'S ALGORITHM	SPECIALIZATION TO THE LDP
<u>Basic Step 3</u> Attempt to find a minimizer Y of $f(X)$ on the linear manifold $A(Q)$. (a) If such a point Y exists it satisfies $f(Y) < f(X^k)$. Go to basic step 4. (b) If f does not have a minimize on $A(Q)$, find its minimizer Y' on A', where A' is the linear manifold through $\hat{X}$ and parallel to $A(Q) \cap A(Q^k)$. Go to basic step 5.	This is identical to step 2 of Wolfe's LDP algorithm. Alternative (b) will never arise in the case of the LDP because $f(X) = X^TX$ will always have a minimizer on the manifold $A(Q)$.
<u>Basic Step 4</u> The barycentric representation of Y on $A(Q)$ is $Y = Qw^*$, where the columns of Q are certain extreme points of the feasible region. There are two cases:	In the case of the LDP, we have $Q = \{P_1, P_2, \ldots, P_j\}$, and thus $Y = Qw^* = P_1w_1^* + P_2w_2^* + \cdots + P_jw_j^*$. Note that $w_j^* > 0$.

VON HOHENBALKEN'S ALGORITHM	SPECIALIZATION TO THE LDP
(a) If $w_i^* > 0$ for all $i = 1,\ldots,j$, then Y belongs to the relative interior of $C(Q)$ and minimizes f on $C(Q)$. Set $X^{k+1} = Y$, $Q^{k+1} = Q$ and go to basic step 1.	This is identical to step 2(C) of Wolfe's LDP algorithm (the feasibility test).
(b) If $w_i^* \leq 0$ for some i, $i \neq j$, then $Y \notin$ rel int $C(Q)$. Go to basic step 5.	
<u>Basic Step 5</u>	
Intersect the boundary of $C(Q)$ with the line segment $\overline{X^kY}$. Let the intersection point be $Z = Q\tilde{w}$.	In the case of the LDP, we may write $Z = P_1\tilde{w}_1 + P_2\tilde{w}_2 + \cdots + P_j\tilde{w}_j$.
The point Z satisfies $f(Z) < f(X^k)$, and will have $\tilde{w}_i \geq 0$, all i, with $\tilde{w}_j > 0$ and at least one $\tilde{w}_i = 0$ for $i \neq j$. <u>Drop</u> the vertices from Q that have $\tilde{w}_i = 0$, to get a reduced affinely independent set $\tilde{Q}$.	This step is identical to step 3 of Wolfe's LDP algorithm, provided the stipulation is made in von Hohenbalken's method that only one point may be dropped from Q during basic step 5.

VON HOHENBALKEN'S ALGORITHM	SPECIALIZATION TO THE LDP
There are two cases: (a) If $C(\tilde{Q})$ is zero-dimensional (i.e. contains only a single point), set $X^{k+1} = Z$, $Q^{k+1} = Q$, and go to basic step 1. (b) If $C(\tilde{Q})$ has positive dimension, set $X^k = Z$, $Q = \tilde{Q}$, go to basic step 3.	Assume that case (a) occurs, i.e. $C(Q)$ consists only of the point Z. von Hohenbalken's method will return to basic step 1 and test point Z for optimality. Wolfe's method, on the other hand, will first determine that Z is the point of smallest norm in $A(\tilde{Q})$ (since $\tilde{Q}$ is a singleton this is the only possibility), and then return to step 1 for the optimality test. It is clear that the steps taken by the two methods are identical. The same holds true if case (b) occurs.

The preceding observations may be summarized as follows:

Proposition 5.1. Suppose that von Hohenbalken's algorithm is applied to the LDP, with the following stipulations:

(1) The starting point is chosen to be a point of P of minimal norm, and

(2) In basic step 5, only one point at a time may be deleted from the set Q.

Then every step taken by the algorithm is identical to the corresponding step of Wolfe's LDP method.

Remark. It follows that von Hohenbalken's algorithm generates the same sequence of primal feasible points as does the van de Panne-Whinston method applied to the LDP. The extent to which this observation may be generalized to quadratic programs of a more general nature is under investigation.

REFERENCES

[1] R.W. Cottle, "The principal pivoting method of quadratic programming," in Mathematics of the Decision Sciences, Part I, Eds. G.B. Dantzig and A.F. Veinott, Jr., Lectures in Applied Mathematics, 11, American Mathematical Society, Providence, R.I. (1968), 144-162.

[2] R.W. Cottle and G.B. Dantzig, "Complementary pivot theory of mathematical programming," in Mathematics of the Decision Sciences, Part I, Eds. G.B. Dantzig and A.F. Veinott, Jr., Lectures in Applied Mathematics, 11, American Mathematical Society, Providence, R.I. (1968), 115-136.

[3] G.B. Dantzig, Linear Programming and Extensions, Princeton University Press, Princeton, N.J. (1963).

[4] E.L. Keller, "Quadratic optimization and linear complementarity," Dissertation, University of Michigan, Ann Arbor, Mich. (1969).

[5] E.L. Keller, "The general quadratic optimization problem," Mathematical Programming 5 (1973), 331-337.

[6] T.D. Parsons, "A combinatorial approach to convex quadratic programming," Dissertation, Princeton University, Princeton, N.J. (1966).

[7] A.W. Tucker, "A least distance approach to quadratic programming," in Mathematics of the Decision Sciences, Part I, Eds. G.B. Dantzig and A.F. Veinott, Jr. Lectures in Applied Mathematics, 11, American Mathematical Society, Providence, R.I. (1968), 163-176.

[8] C. van de Panne and A. Whinston, "The symmetric formulation of the simplex method for quadratic programming," Econometrica 37 (1969), 507-527.

[9] B. von Hohenbalken, "A finite algorithm to maximize certain pseudoconcave functions on polytopes," Mathematical Programming 9 (1975), 189-206.

[10] D.R. Wilhelmsen, "A nearest point algorithm for convex polyhedral cones and applications to positive linear approximation," Math. of Comp. 30 (1976), 48-57.

[11] P. Wolfe, "Algorithm for a least-distance programming problem," Mathematical Programming Study I (North Holland) (1974), 190-205.

[12] P. Wolfe, "Finding the nearest point in a polytope," Report RC 4887, IBM Research Center, Yorktown Heights, N.Y. (1974).

[13] G. Zoutendijk, Methods of Feasible Directions, Elsevier, Amsterdam (1960).

(14) TR-76-26

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 76-26

2. GOVT ACCESSION NO.:

3. RECIPIENT'S CATALOG NUMBER:

4. TITLE (and Subtitle): (6) ALGORITHMIC EQUIVALENCE IN QUADRATIC PROGRAMMING I: A LEAST-DISTANCE PROGRAMMING PROBLEM

5. TYPE OF REPORT & PERIOD COVERED: (9) Technical Report

6. PERFORMING ORG. REPORT NUMBER:

7. AUTHOR(s): (10) Richard W. COTTLE and Arthur DJANG

8. CONTRACT OR GRANT NUMBER(s): (15) N00014-75-C-0267, NSF-MCS-71-03341

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Department of Operations Research
Stanford University
Stanford, CA 94305

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: NR-047-064

11. CONTROLLING OFFICE NAME AND ADDRESS:
Operations Research Program
Office of Naval Research
Arlington, VA 22217

12. REPORT DATE: (11) October 1976

13. NUMBER OF PAGES: 64 (12) 68p.

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):

15. SECURITY CLASS. (of this report):

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report):
Approved for public release and sale; distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):

18. SUPPLEMENTARY NOTES:

19. KEY WORDS (Continue on reverse side if necessary and identify by block number):
ALGORITHMS
EQUIVALENCE
LEAST DISTANCE PROBLEM
QUADRATIC PROGRAMMING

20. ABSTRACT (Continue on reverse side if necessary and identify by block number):

It is demonstrated that Wolfe's algorithm for finding the point of smallest Euclidean norm in a given convex polytope generates the same sequence of feasible points as does the van de Panne-Whinston "symmetric" algorithm applied to the associated quadratic programming problem. Furthermore, it is shown how the latter algorithm may be simplified for application to problems of this type.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

402766